

# Faxina

## Introdução à Faxina de dados



# Introdução

# O que é faxina?

**Faxina de dados** é o processo de modificar uma ou mais tabelas até que elas atendam o princípio **tidy**, proposto por Hadley Wickham em um artigo de 2014.

Cada base exigirá uma **Faxina** diferente, pois não existe um único tipo de bagunça:

*Tidy data ~~Happy families~~ are all alike; every untidy data ~~unhappy family~~ is untidy ~~unhappy~~ in its own way (Hadley Wickham ~~Leon Tolstoi~~)*

Nesse curso, vamos cobrir algumas bagunças comuns e exemplos práticos de como resolvê-las.

# Por que faxina é importante?

Dados são o subproduto de muitos processos complexos, administrado por várias pessoas que podem fazer (ou não) usos muito diferentes dos mesmos registros.

Isso faz com que os formatos e convenções de armazenamento de dados sejam muito variados, às vezes dificultando chegando a dificultar análises futuras.

Por isso, a Faxina é praticamente uma constante em qualquer contexto de análise de dados, e isso não se deve **só** a erros ou falhas no processo de armazenamento.

Bancos de dados são usualmente armazenados em formato relacional (não *tidy*), por exemplo, pois isso reduz o espaço consumido pelas informações.

# O que é um banco de dados tidy

*Tidy data* é um princípio para arrumação de base de dados que resolve 90% dos problemas reais. Uma base tidy satisfaz é **uma única tabela** que satisfaz:

- Cada observação é uma (e só uma) linha da tabela.
- Cada variável é uma coluna da tabela. Não existe uma coluna que represente duas variáveis, por exemplo.

Essas definições são interdependentes, a depender do que entendermos por **variável** e **observação**. O que realmente é a filosofia por trás das sugestões.

Essa é a definição original proposta anteriormente, mas neste curso vamos considerar algumas propriedades adicionais. Elas não são tão precisas, mas se justificam pela sua grande aplicabilidade:

- Não existem colunas com tipos trocados: datas ou números como texto.
- As lacunas da base são lacunas verdadeiras. Não existe um "" que na verdade é um valor faltante.

# Relação com outros formatos de armazenamento

O contrário **tidy** não é **messy**. Existem muitos motivos pelos quais você se deparar com dados fora do padrão que vamos adotar neste curso. Os dois principais são:

1. Dados em formato **tidy** não otimizam o uso de espaço no computador. Frequentemente nós acabamos criando linhas novas para deixar uma tabela **tidy** e isso pode acabar fazendo o computador armazenar colunas muito maiores.
2. Dados em formato **tidy** podem não ser os melhores para fazer consultas. O formato **tidy** nos ajuda a gastar menos tempo fazendo as nossas análises, mas pode dificultar a vida de quem precisa consultar os mesmos dados que vamos analisar.

Por esses dois motivos, lembre-se: o contrário de **tidy** não é **messy**. Em muitas situações falta cuidado com o armazenamento dos dados e de fato podemos falar em **BAGUNÇA**, mas em muitas outras os dados são **untidy** por algum bom motivo.

# Exemplo de dado untidy

Casos de tuberculose em três países de 1999 a 2000:

pais	1999	2000
Afeganistão	745	2666
Brasil	37737	80488
China	212258	213766

Porque não é **tidy**:

- As colunas "1999", "2000" representam uma variável numérica (**casos**), mas isso não aparece na tabela.
- As **observações** são 6: uma para cada país em cada ano. As qualificações dessas observações, o **ano** e o **país** em que elas foram feitas, não aparecem na base: falta uma coluna com os **casos**

# Versão tidy

país	ano	casos
Afeganistão	1999	745
Afeganistão	2000	2666
Brasil	1999	37737
Brasil	2000	80488
China	1999	212258
China	2000	213766



# Exemplo de dado untidy

Casos de tuberculose em três países de 1999 a 2000:

pais	1999	2000
Afeganistão	745	2666
Brasil	37737	80488
China	212258	213766

Porque não é **tidy**:

População em três países de 1999 a 2000:

pais	1999	2000
Afeganistão	19987071	20595360
Brasil	172006362	174504898
China	1272915272	1280428583

- Além dos problemas anteriores, as informações estão espalhadas em várias tabelas.

# Versão tidy

<b>país</b>	<b>ano</b>	<b>casos</b>	<b>populacao</b>
Afeganistão	1999	745	19987071
Afeganistão	2000	2666	20595360
Brasil	1999	37737	172006362
Brasil	2000	80488	174504898
China	1999	212258	1272915272
China	2000	213766	1280428583

# Tipos de problemas mais comuns

Erro	Solução	Função no R
Uma variável separada em várias colunas	Transformar várias colunas em uma e empilhar linhas	<code>tidyr::pivot_longer</code>
Informação espalhada em várias tabelas	Juntas as tabelas pelo valor de uma coluna	<code>dplyr::xxx_join</code>
Variáveis diferentes empilhadas como linhas	Transformar as linhas repetidas em colunas	<code>tidyr::pivot_wider</code>
Uma tabela fruto de muitos joins	Separar as subtabelas e transformar informações repetidas em uma única cela	<code>tidyr::nest()</code> - <code>tidyr::unnest()</code> - <code>dplyr::group_by()</code> + <code>dplyr::summarise()</code>

# Bases de dados

## Aula 01

- Base com várias tabelas entranhadas
- Base de municípios sem código
  - de-para
  - inconsistências e lacunas
- Montando tabelas com os dados
  - funcoes legais do purrr, tidyr, dplyr Aula 02
- Arquivos PDF serenata de amor
  - pdftools
  - tabulizer
  - extraindo dados de texto
- Preparando base para shiny
  - reduzir tamanho de arquivo
  - nested tibbles Aula 03
- Kaggle base retangular
- Kaggle base de textos

# Vamos ao R!

